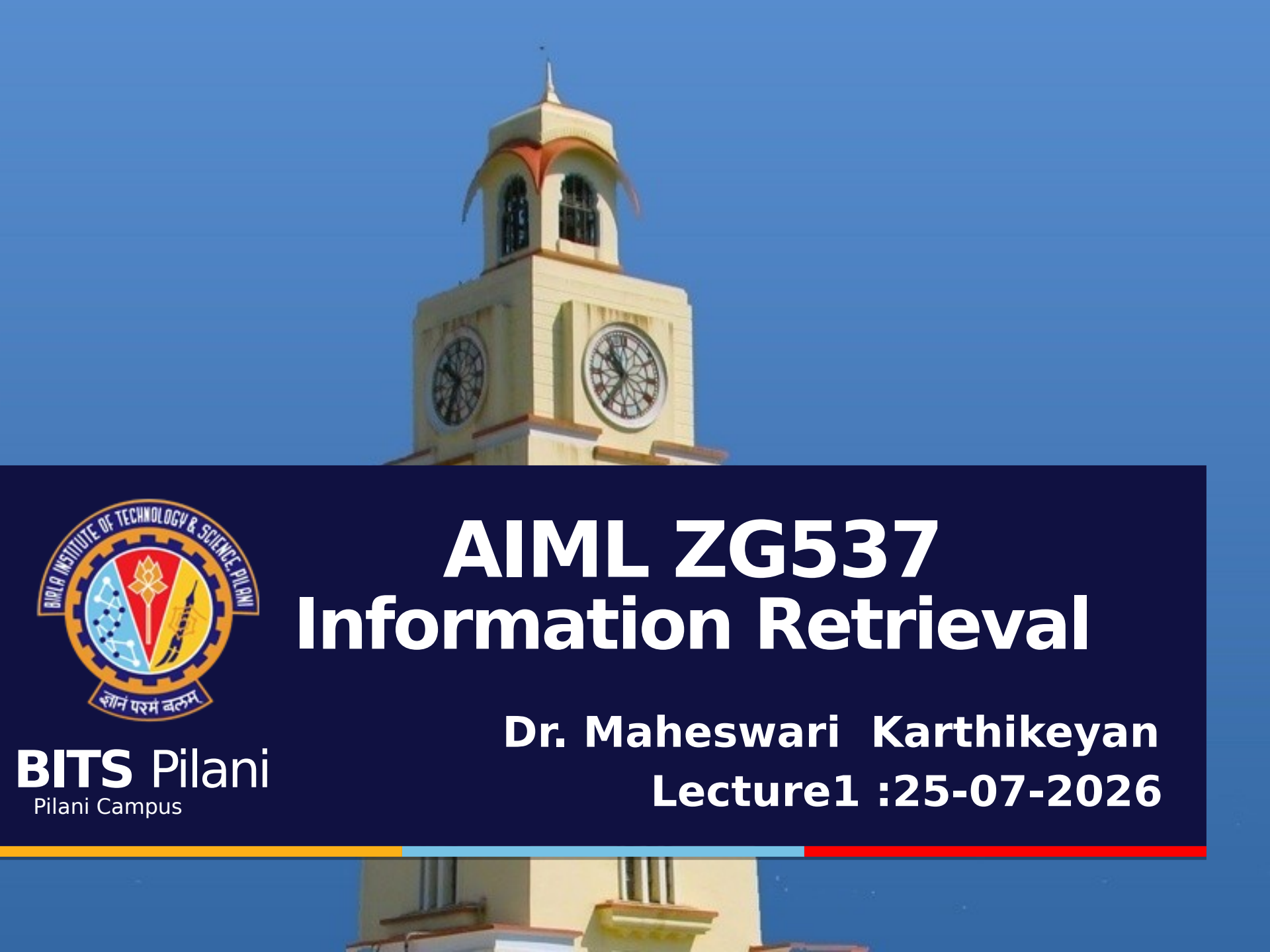


AIML ZG537
Information Retrieval

Dr. Maheswari Karthikeyan

Lecture1 :25-07-2026

BITS Pilani
Pilani Campus

Course Outline

- **To acquire basic understanding of the components and the different IR methods.**
  - Boolean
  - Vector Space
- **To understand the various application areas of IR:**
  - Text Mining
  - Web Search
  - Cross Lingual IR
  - Multimedia IR
  - Recommender System
  - Neural IR

Books to Refer

1. C. D. Manning, P. Raghavan and H. Schutze. Introduction to Information Retrieval, Cambridge University Press, 2008. http://nlp.stanford.edu/IR-book/
2. Modern Information Retrieval, Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Addison-Wesley, 2000. http://people.ischool.berkeley.edu/~hearst/irbook/
3. Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. (Eds.), Recommender Systems Handbook. 1st Edition., 2011, 845 p. 20 illus., Hardcover, ISBN: 978-0-387-85819-7

Lecture Outline

innovate achieve lead

Introduction

- Information Retrieval
- Information vs. Data Retrieval
- IR task
- Basic Concepts
- Logical view of the documents
- The retrieval process
- Classical IR models

Information Retrieval

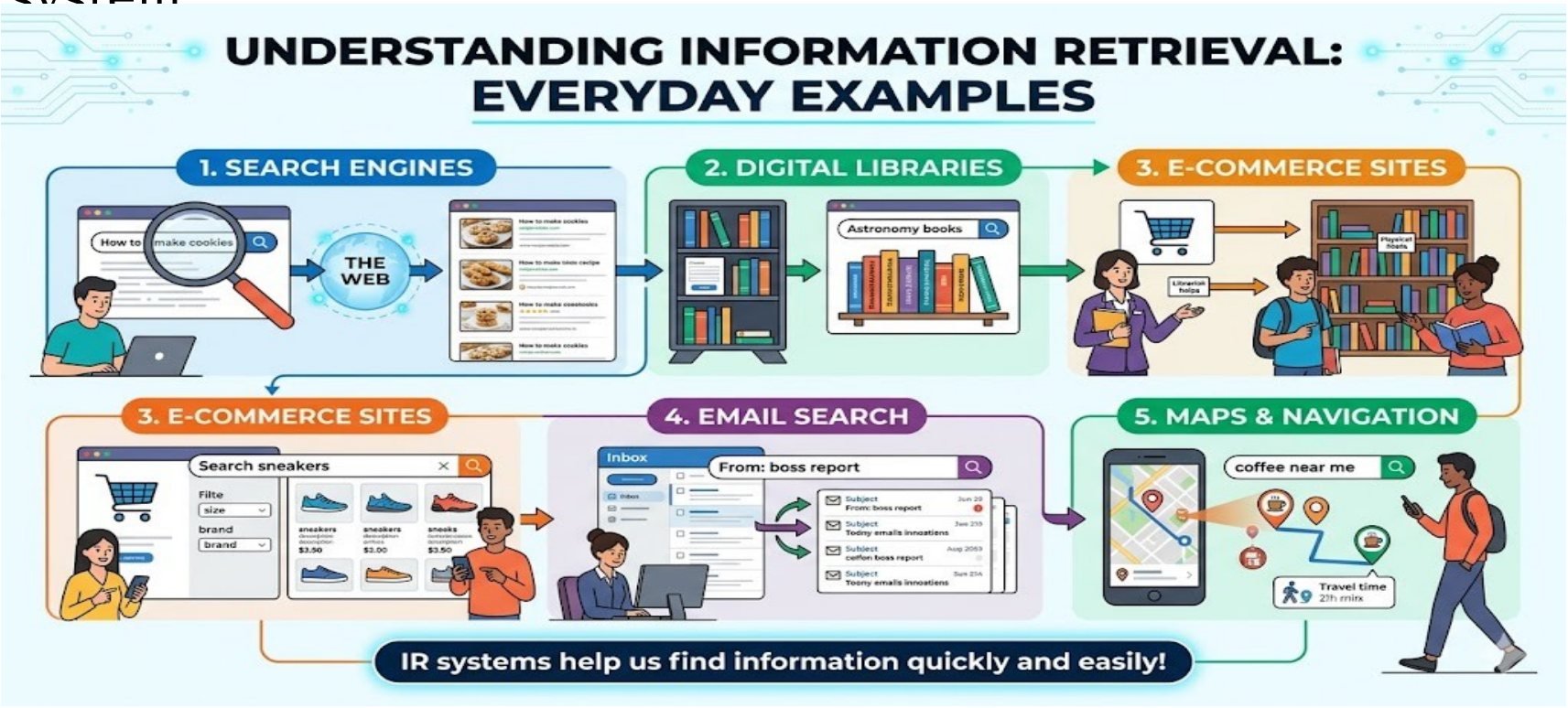
- Information Retrieval (IR) is **finding material** (usually documents) of an **unstructured** nature (usually text) that satisfies an **information need** from within **large collections**.

……Not restricted to Web search

- E-mail search
- Searching laptop
- Corporate knowledge bases
- Legal information retrieval

Applications of Information Retrieval

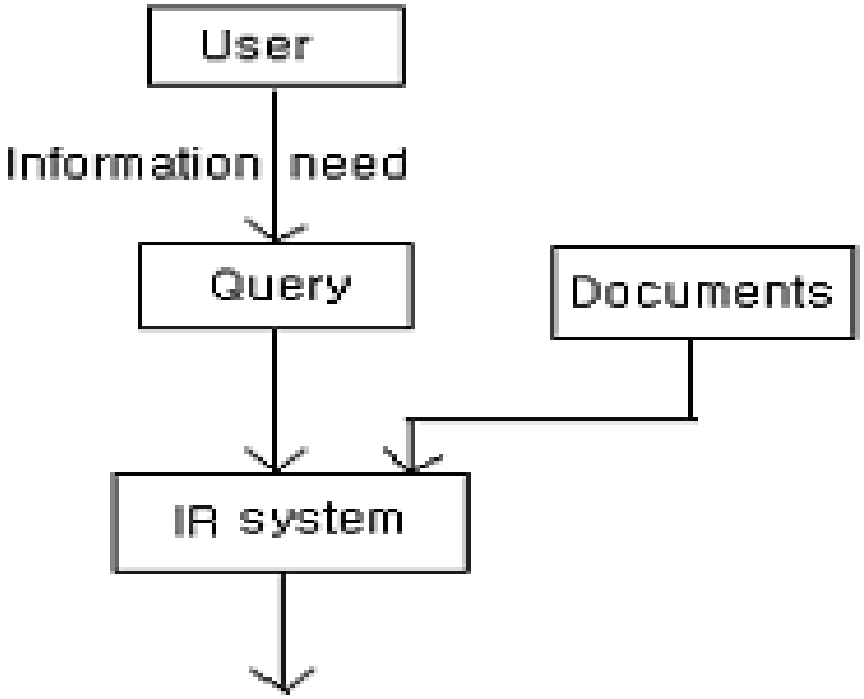
Every time you use a web search engine, search your email, or query a library catalog, you are interacting with an IR system



Applications of Information Retrieval

- Digital Libraries
- Search Engines
- Media search
- Information Filtering
- Legal Information Retrieval
- Document Classification
- Question Answering

Information Retrieval



Basic Information Retrieval Process

Motivation

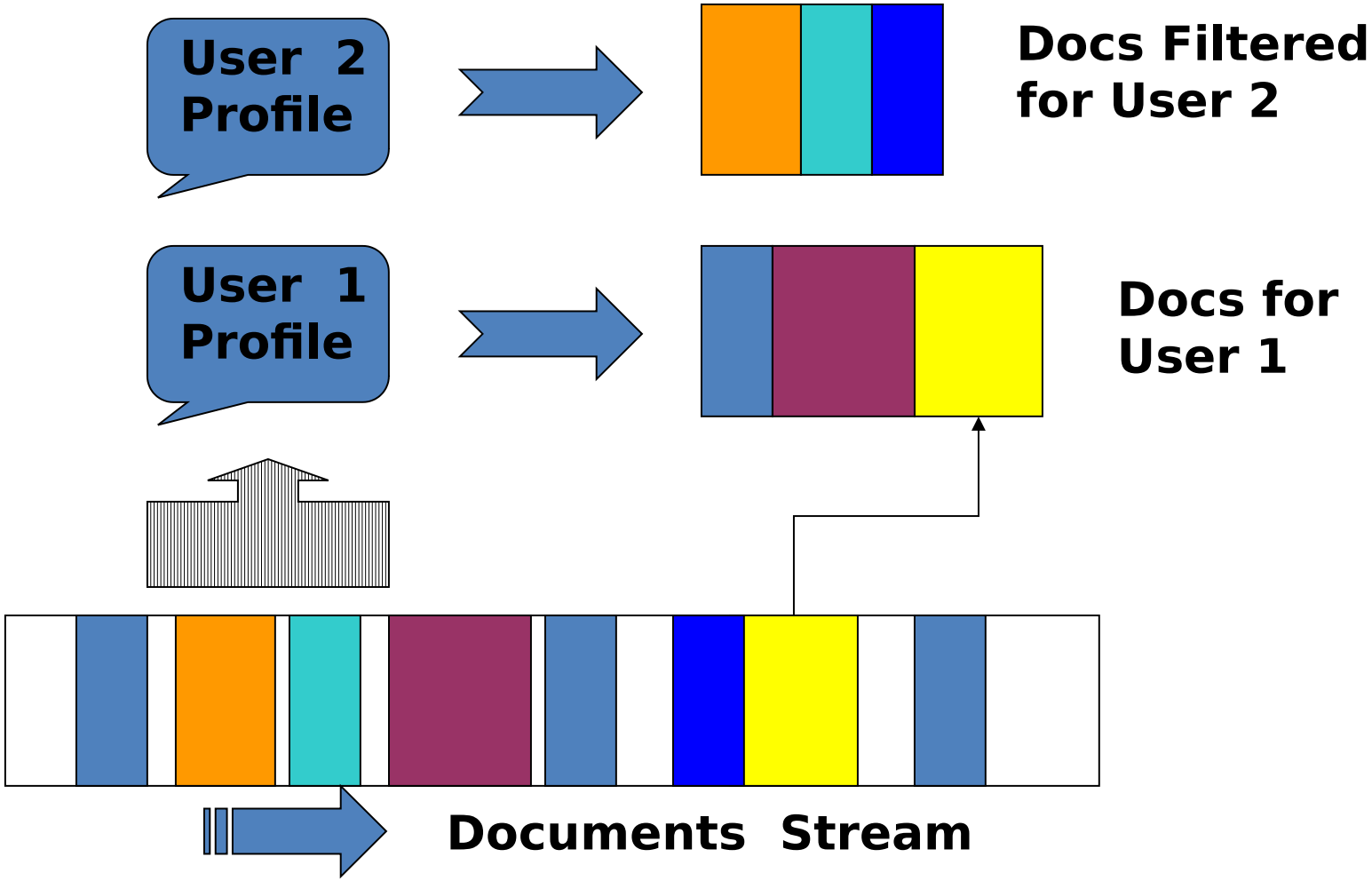
- Information Retrieval (IR) is about:
  - Representation
  - Storage
  - Organization of
  - And access to “information items”
- Focus is on user’s “information need” rather than a precise query.
- Emphasis is on the retrieval of *information* (not data)

Types of Information Needs

- **Retrospective**
  - "Searching the past"
  - Different queries posed against a static collection
  - Time invariant
- **Prospective**
  - "Searching the future"
  - Static query posed against a dynamic collection
  - Time dependent

Retrieval - Ad hoc

Retrieval - Filtering

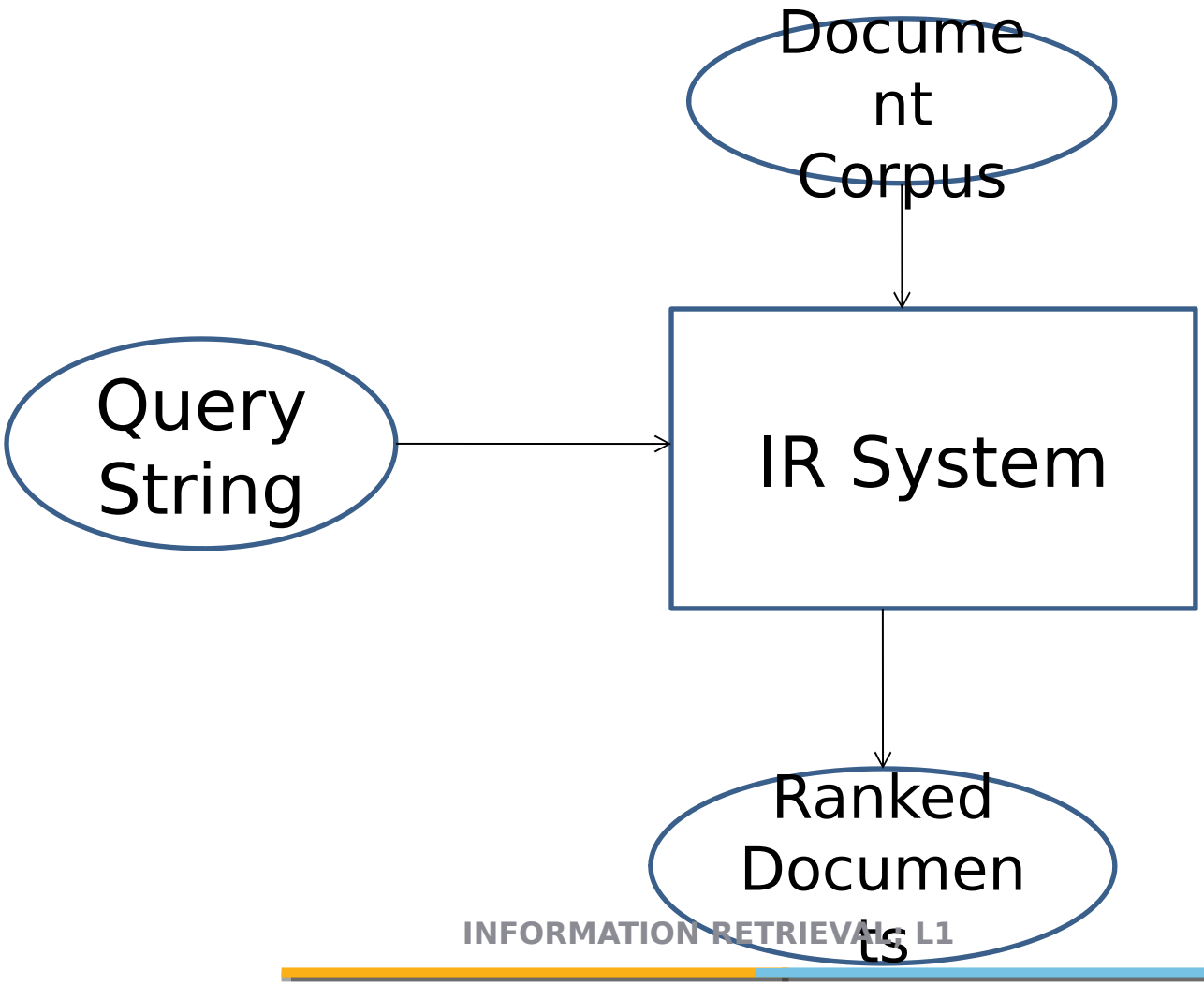


IR Task

innovate achieve lead

- **Input:**
  - A corpus of textual natural-language documents
  - A user query in the form of a textual string
- **Output:**
  - A ranked set of documents that are relevant to the query.

IR Task



Relevance

- Relevance is a subjective judgment and may include:
  - Being on the proper subject.
  - Being timely (recent information).
  - Being authoritative (from a trusted source).
  - Satisfying the goals of the user and intended use of the information (*information need*).

Intelligent IR

- **Meaning of the words** used
- **Order of words** in the query
- **Direct or indirect feedback**
- **Authority** of the source

IR vs. Data Retrieval

- **Data retrieval**
  - Which documents contain a set of keywords?
  - Well defined structure and semantics
  - A single erroneous object implies failure
  - Provide solution to the user of a database system
- **Information retrieval**
  - Information about a subject or topic
  - Semantics is frequently loose
  - Small errors are tolerated
  - Deals with natural language text

IR vs. Data Retrieval

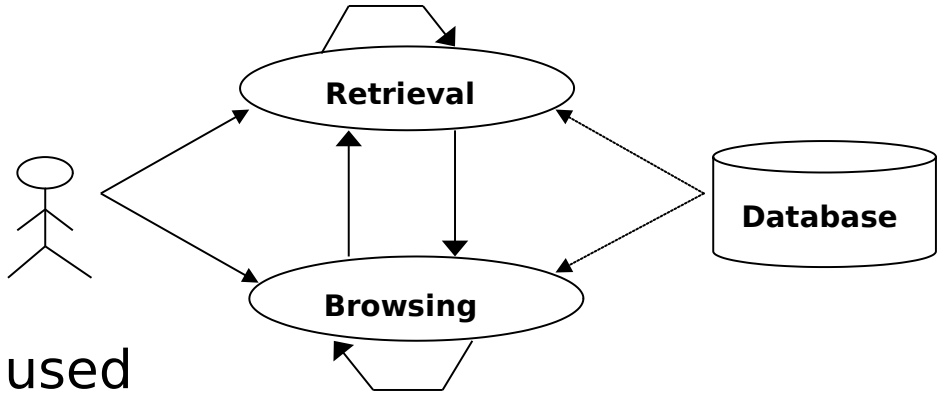
	Data	IR
Data	**Structured**	**Unstructured**
Fields	**Clear semantics** (SSN, age)	**No fields** (other than text)
Queries	**Defined** (relational algebra, SQL)	**Free text** ("natural language"), Boolean
Matching	**Exact** (results are *always* "correct")	**Imprecise** (need to measure effectiveness)

IR System -Basic Concepts

- Efficient retrieval system is directly related to
  - User task
  - Logical view of the documents

User Task

- The User Task
  - Retrieval
    - Information or data
    - Purposeful
  - Browsing
    - Hypertext systems used
    - Glancing around



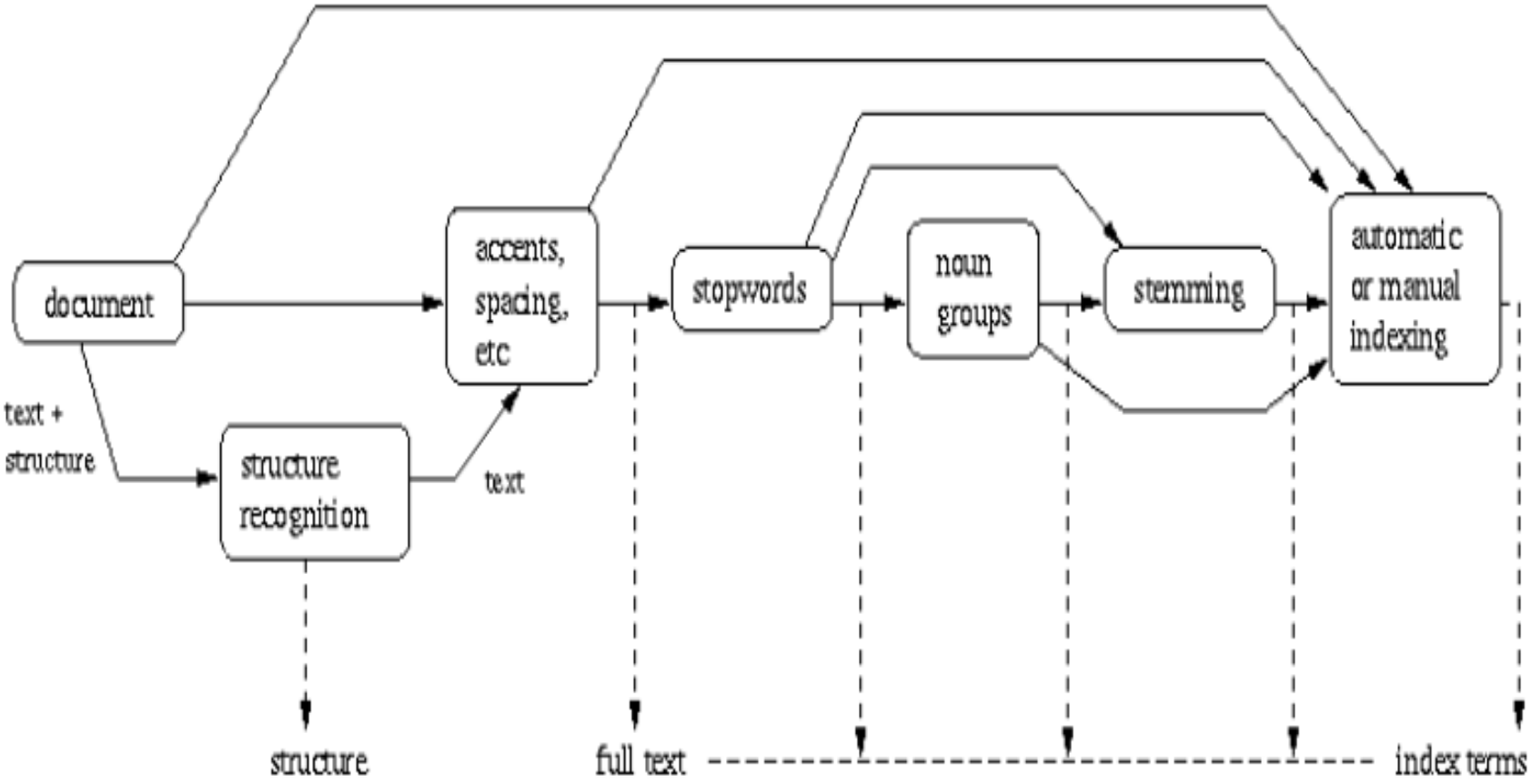
Interaction of the user with the retrieval system through distinct tasks

- Both retrieval(adhoc) and browsing are "pulling" actions
- Alternative is to "push" the information towards the user, to execute the particular retrieval task which consists of "filtering" relevant information.

Logical view of the documents

- Documents in a collection are frequently represented through a set of index terms or keywords
- Keywords are extracted from document
- Keywords are derived automatically or generated by a specialist, they provide a logical view of the document
- Stop-words
  - To reduce the set of representative keywords from large collection
- Function words do not bear useful information for IR,
  - i.e. of, in, about, with, I, although, …
- Stop-list: contain stop-words, not to be used as index
  - Prepositions, Articles, Pronouns
  - Some adverbs and adjectives, Some frequent words (e.g. document)
- The removal of stop-words usually improves IR effectiveness
- A few “standard” stop-lists are commonly used.

Logical View of the Document



Logical view of the document: from full text to a set of index terms

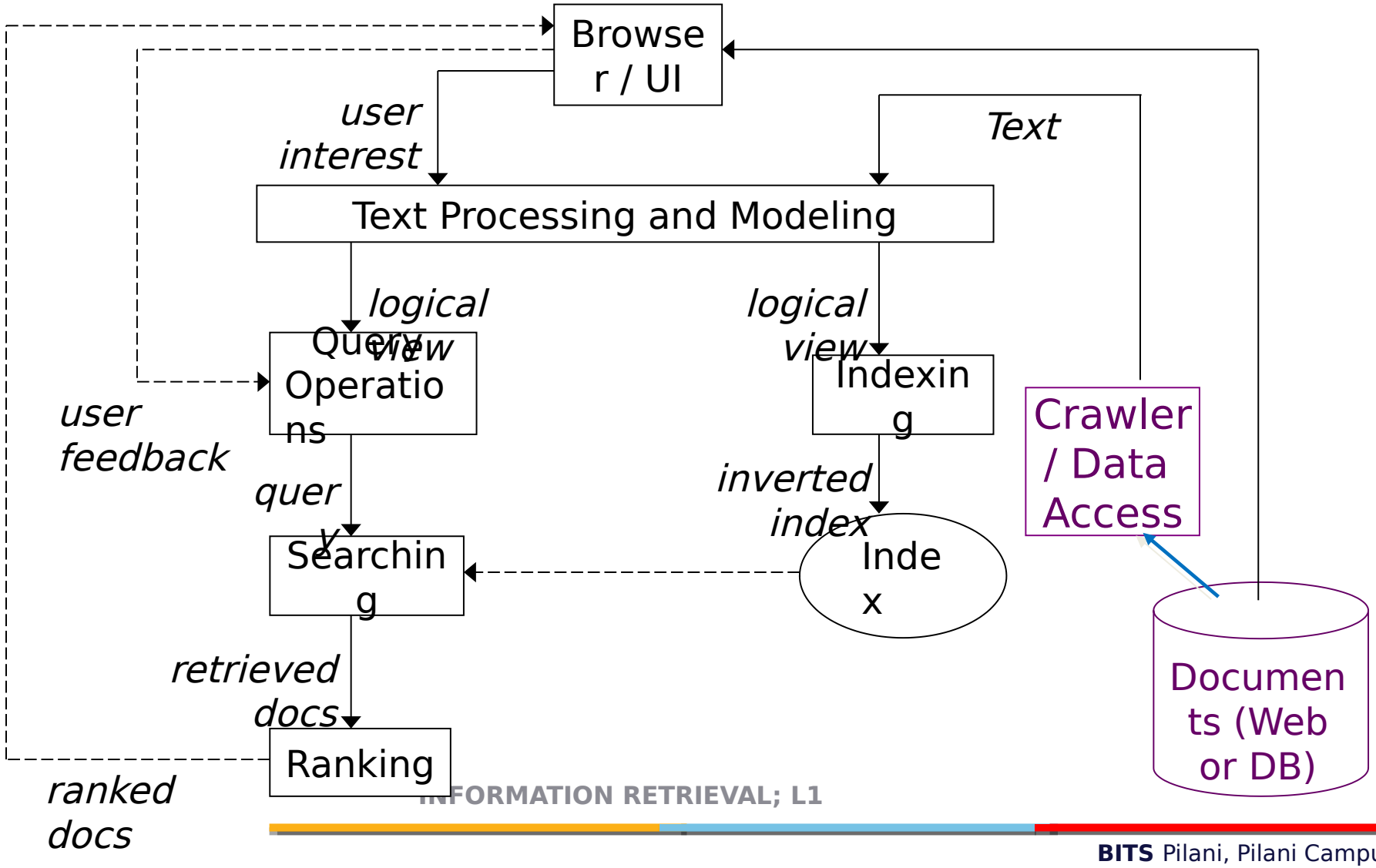
Logical view of the documents

- **Noun groups**
  - To identify the noun groups
  - Which eliminates the adjectives, adverbs and verbs
- **Reason for stemming**
  - Different word forms may bear similar meaning (e.g. search, searching): create a “standard” representation for them
- **Stemming**
  - Which reduces distinct words to their common grammatical root
  - Removing some endings of word

computer
compute
computes
computing
computed
computation

→ **comput**

The Retrieval Process



IR System Components

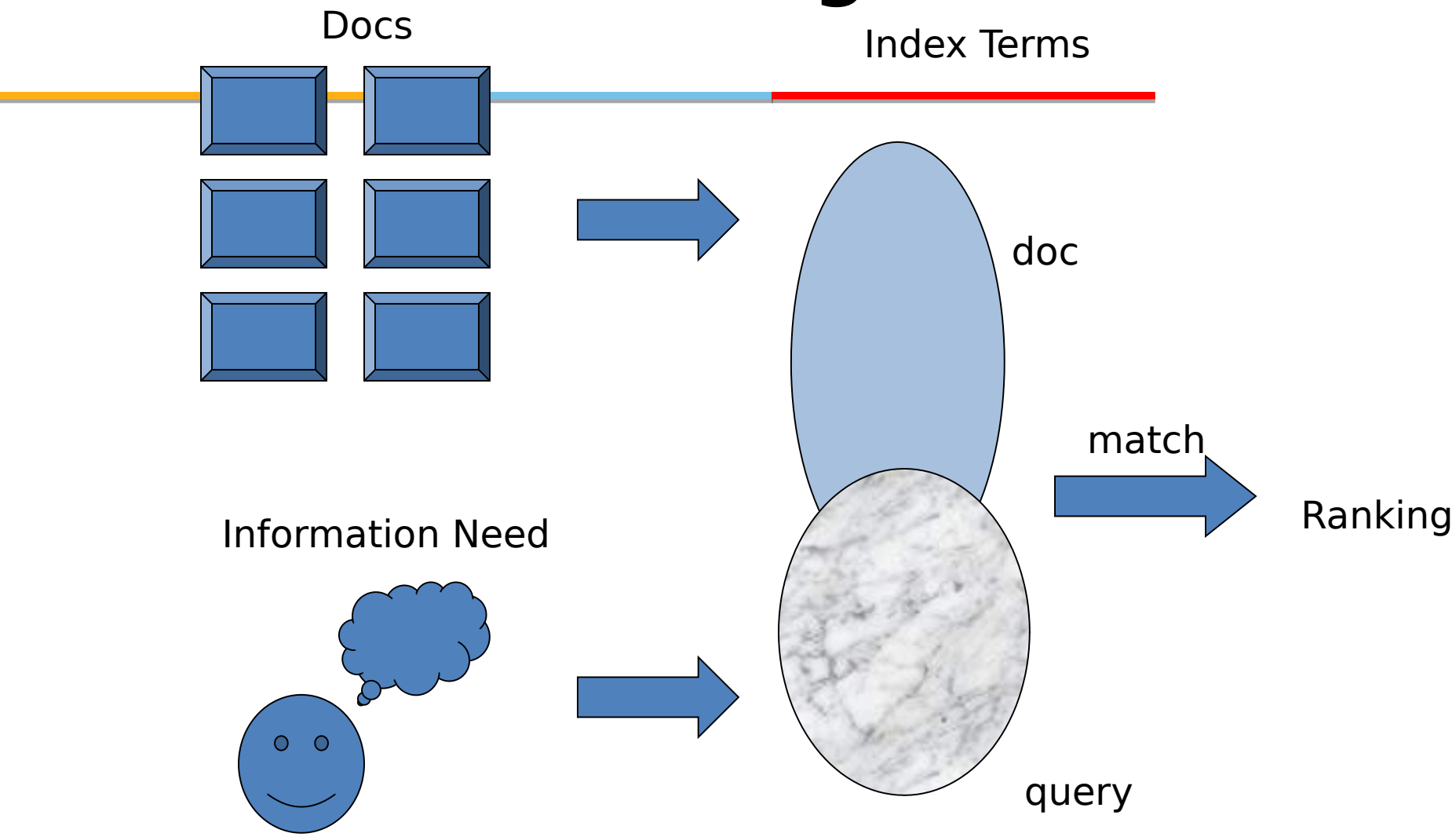
- Text Operations forms index words (tokens)
  - Stop-word removal
  - Stemming
- Indexing constructs an *inverted index* of word to document pointers
- Searching retrieves documents that contain a given query token from the inverted index
- Ranking scores all retrieved documents according to a relevance metric
- User Interface manages interaction with the user:
  - Query input and document output.
  - Relevance feedback.
  - Visualization of results.
- Query Operations transform the query to improve retrieval:
  - Query expansion
  - Query transformation using relevance feedback

Information Retrieval Models

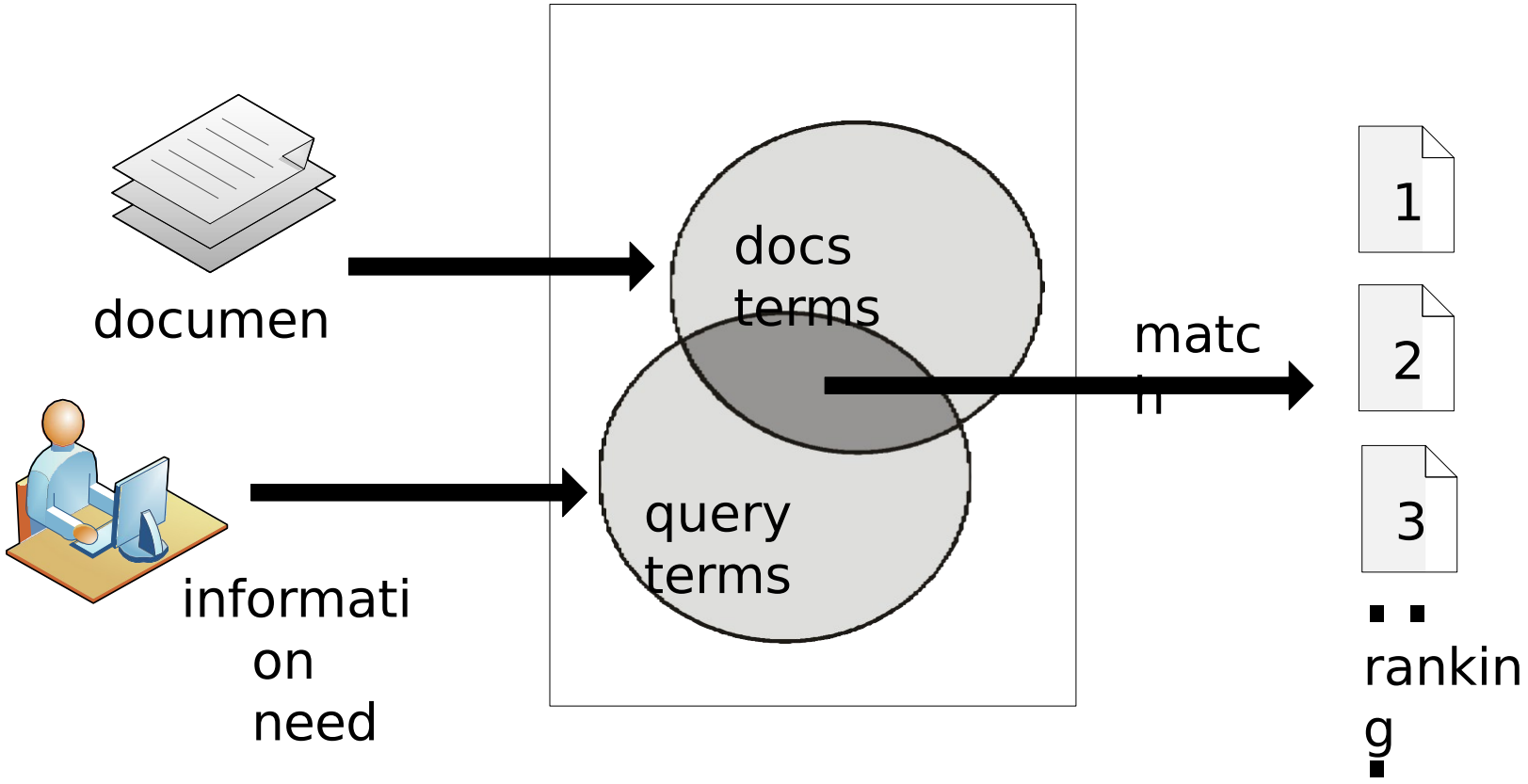
Information Retrieval Models

- Traditional IR uses *Index Terms* to retrieve documents
- A *ranking* is an ordering of the documents retrieved to the user query
- A ranking is based on fundamental premises regarding the notion of relevance, such as:
  - common sets of index terms
  - sharing of weighted terms
  - likelihood of relevance
- Each set of premises leads to a distinct *IR model*

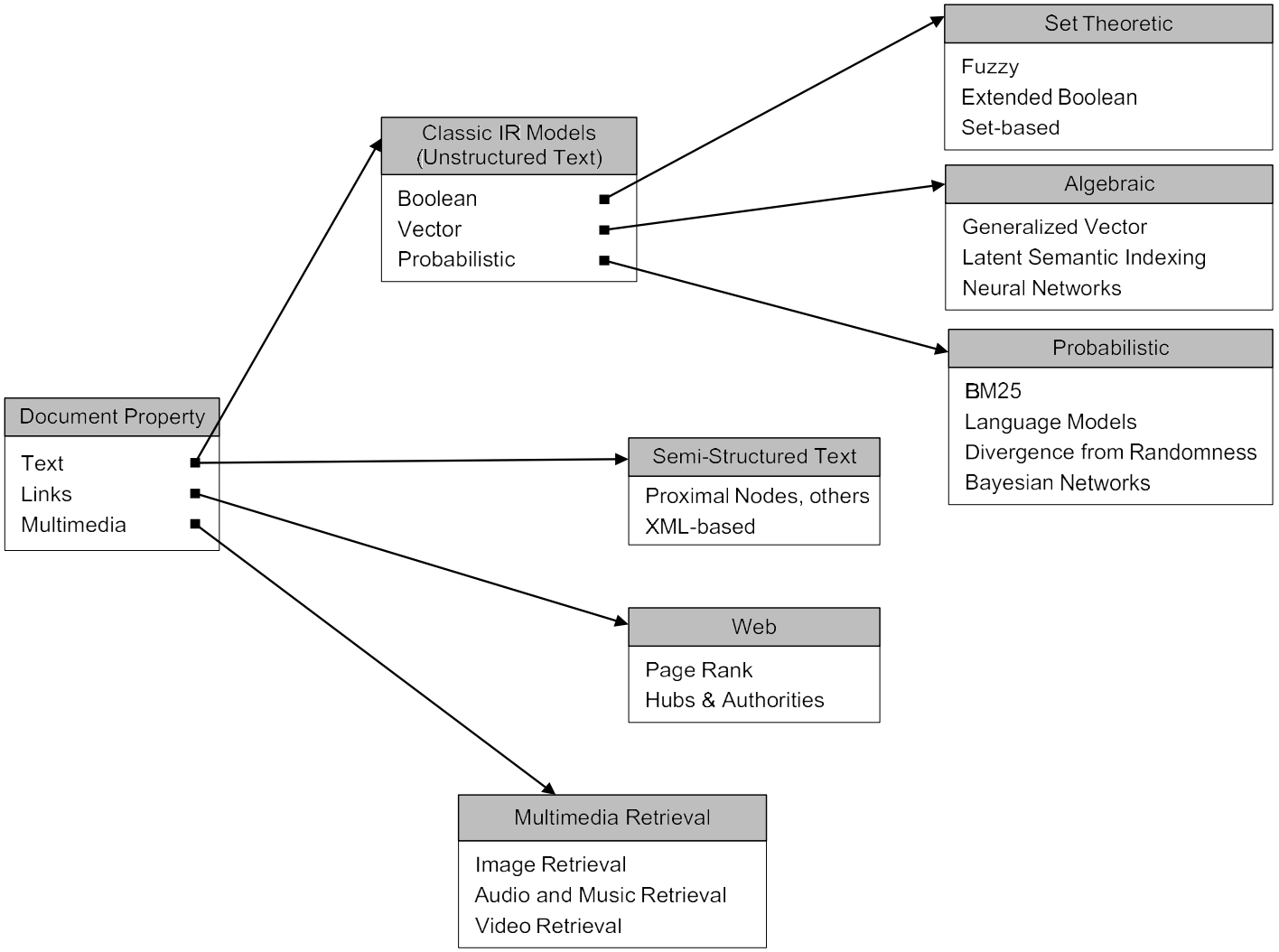
Modeling



Modeling


documen
informati
on
need
docs
terms
query
terms
matc
h
1
2
3
rankin
g

Taxonomy of IR Models



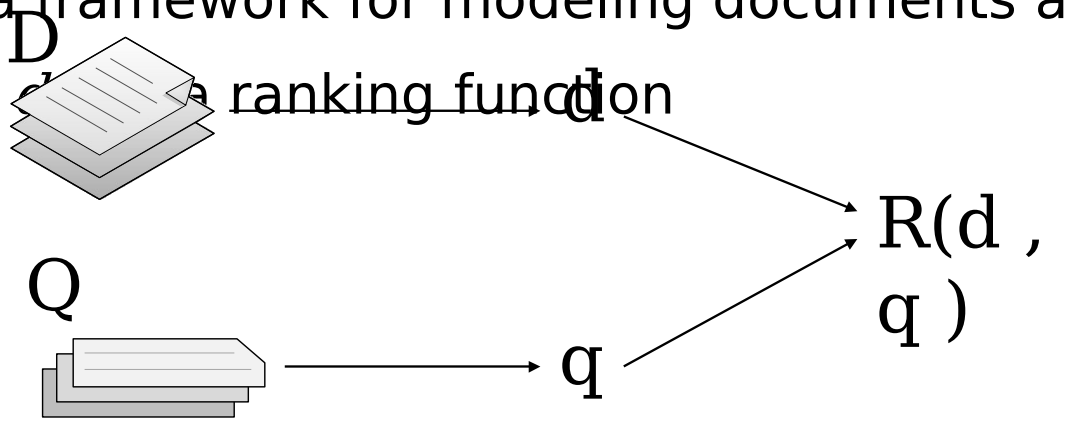
Classic IR Models – Basic Concepts

- Each document represented by a set of representative keywords or index terms
- An index term is a document word useful for remembering the document main themes
- Traditionally, index terms were nouns because nouns have meaning by themselves
- However, search engines assume that all words are index terms (full text representation)

Classic IR Models – Basic Concepts

An **IR model** is a quadruple [D, Q, F, $R(q_i, d_j)$] where

1. D is a set of logical views for the documents in the collection
2. Q is a set of logical views for the user queries
3. F is a framework for modeling documents and queries
4. $R(q_i, d_j)$ is a ranking function



Classic IR Models – Ranking

- Not all terms are equally useful for representing the document contents: less frequent terms allow identifying a narrower set of documents
- The *importance* of the index terms is represented by weights associated to them
- Let
  - k_i be an index term
  - d_j be a document
  - w_{ij} is a weight associated with (k_i, d_j)
- The weight w_{ij} quantifies the importance of the index term for describing the document contents

Classic IR Models – Notations

k_i is an index term (keyword)

d_j is a document

t is the total number of docs

$K = (k_1, k_2, \ldots, k_t)$ is the set of all index terms

$w_{ij} >= 0$ is a weight associated with (k_i, d_j)

$w_{ij} = 0$ indicates that term does not belong to doc

$vec(d_j) = (w_{1j}, w_{2j}, \ldots, w_{tj})$ is a weighted vector associated with the document d_j

$g_i(vec(d_j)) = w_{ij}$ is a function which returns the weight associated with pair (k_i, d_j)

Classical IR Models

- Boolean model
- Vector Space model
- Probabilistic model

Boolean Model

- Simple model based on set theory and Boolean algebra
  - Documents are sets of terms
  - Queries are Boolean expressions on terms
- Historically the most common model
  - Library OPACs
  - Dialog system
  - Many web search engines
- Queries specified as boolean expressions
  - Precise semantics
  - Neat formalism
- Terms are either present or absent. Thus, $w_{ij} \varepsilon \{1,0\}$

Boolean Model

- **D: set of words (indexing terms) present in a document**
  - each term is either present (1) or absent (0)
- **Q: A Boolean expression**
  - terms are index terms
  - operators are AND, OR, and NOT
- **F: Boolean algebra over sets of terms and sets of documents**
- **R: a document is predicted as relevant to a query expression if it *satisfies the query expression***
- **((*text* ˅ *information)* ˄ *retrieval* ˄ ¬*theory)***
- Each query term specifies a set of documents containing the term
- AND (˄): the intersection of two sets
- OR (˅): the union of two sets
- NOT (¬): set inverse, or really set difference

Boolean Model

- **Advantages**
  - Clean Formalism
  - Easy to implement
  - Intuitive concept
  - Still, it is a dominant model for document database systems.

Limitations of Boolean Model

- Retrieval based on binary decision criteria with no notion of partial matching
- No ranking of the documents is provided (absence of a grading scale)
- Information need has to be translated into a Boolean expression which most users find difficult
- The Boolean queries formulated by the users are most often too simplistic
- Frequently returns either too few or too many documents in response to a user query.

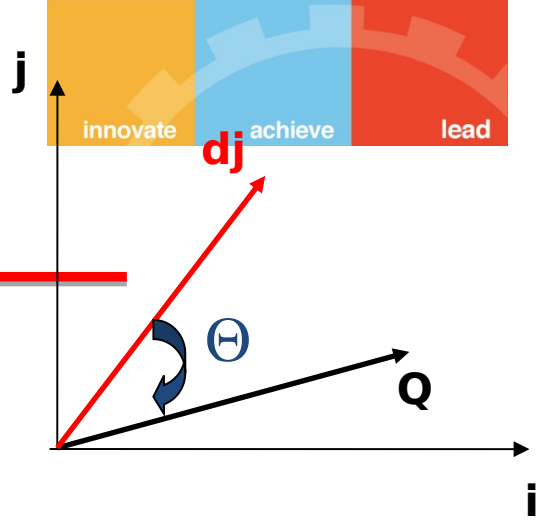
Vector Model

- Use of binary weights is too limiting
- Non-binary weights provide consideration for partial matches
- These term weights are used to compute a *degree of similarity* between a query and each document
- Ranked set of documents provides for better matching

Define:

- $w_{i,j} >= 0$ associated with the pair (ki, dj) $\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$
- $vec(d_j) = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$
- $w_{i,q} >= 0$ associated with the pair (k_i, q) $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$
- $vec(q) = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$
- *t- total no. of index terms in the collection*

Vector Model



$Sim(d_j, q) =$

- A document is retrieved even if it matches the query terms only partially

A good weight must take into account of two effects:

- quantification of intra-document contents (similarity)
  - *tf* factor, the *term frequency* within a document
- quantification of inter-documents separation (dis-similarity)
  - *idf* factor, the *inverse document frequency*
- $w_{ij} = tf * idf$

Vector Model

- **Advantages**
  - Simple model based on linear algebra
  - Term weights not binary
  - Allows computing a continuous degree of similarity between queries and documents
  - Allows ranking documents according to their possible relevance
  - Allows partial matching
  - Allows efficient implementation for large document collections

Vector Model

- **Disadvantages**
  - Index terms are assumed to be mutually independent
  - Search keywords must precisely match document terms
  - Long documents are poorly represented
  - The order in which the terms appear in the document is lost in the vector space representation
  - Weighting is intuitive, but not very formal.

Probabilistic model

- The model is called as BIR (Binary Independence Retrieval)
- It uses a probabilistic framework
- Given a user query, there is an *ideal* answer set
- Guess at the beginning what they could be (i.e., guess initial description of ideal answer set)
- User look retrieved doc's are either relevant or non-relevant
- Improve by iteration.

Probabilistic model

- An initial set of documents is retrieved, can be done using vector model, Boolean model
- User inspects these docs looking for the relevant ones
- IR system uses this information to refine description of ideal answer set
- By repeating this process, it is expected that the description of the ideal answer set will improve
- Description of ideal answer set is modelled in probabilistic terms.

Probabilistic model- Ranking

- Given a user query *q* and a document *dj*, the probabilistic model tries to estimate the probability that the user will find the document $\boldsymbol{d_j}$ interesting (i.e., relevant)
- The model assumes that this probability of relevance depends on the query and the document representations only
- Ideal answer set is referred to as *R* and should maximize the probability of relevance. Documents in the set *R* are predicted to be relevant.

Probabilistic model

- **Advantages**
  - Documents are ranked in decreasing order of their probability of relevant
- **Disadvantages**
  - Need to guess the initial separation of documents into relevant and non-relevant sets
  - All weights are binary
  - The adoption of the independence assumption for index terms.

Resources

Xapian

OpenSearch

Resources

https://ieeexplore.ieee.org/document/10184013

Information Retrieval: Recent Advances and Beyond
KAILASH A. HAMBARDE AND HUGO PROENÇA , (Senior Member, IEEE)